METHOD TO INVOKE WIDE-AREA OBJECTS IN DISTRIBUTED COMPUTER SYSTEMS

TECHNICAL FIELD

The present invention relates generally to distributed computer systems that consist of a number of software objects that reside in a number of data centers and more specifically a fault-tolerant method to invoke wide-area objects.

BACKGROUND ART

In the past, there have been distributed systems that consist of a number of software objects that reside in a number of data centers. The software objects can be replicated databases, or other types of systems. A local-area network, such as an Ethernet, mediates communication between objects in the same data center. Communication between objects that reside in different data centers takes place via a wide-area network, such as a leased phone line. The dispersion of objects across multiple data centers allows a system to be resilient to disasters that cause a data center to go down. The multiplicity of objects within a data center makes each data center fault- tolerant: a data center can continue to deliver its intended function even if some of its objects fail.

The scenario is the following: a given object, called the initiator, wants to invoke a given method in all objects. It is necessary that objects be invoked reliably: informally, the failure of an object should not prevent other (correct) objects from being invoked. The invocation protocol should be efficient: since data centers are connected to each other via wide-area networks, and since such networks are slow and unpredictable, it is desirable to minimize the communication between data centers without compromising the reliability of the system.

There are existing solutions for so-called reliable broadcast. One common way to implement reliable broadcast is message diffusion. With message diffusion, the basic idea is that any receiver of a broadcast message relays the message to all other objects in the system. With this scheme, all correct processes eventually receive the broadcast message. The problem with message diffusion is that any correct object will propagate each message to all other objects, which means that the number of messages communicated across wide-area links is proportional to the square of the number of objects.

Another way to implement reliable broadcast is to use failure detection. If a first object receives a message from a second object, the following takes place. If the first object does not suspect the second object to have failed it does nothing. If the first object suspects the second object to have failed it relays the message to the other objects in the system. The number of message communicated across wide-area links is here proportional to the number of objects.

A protocol (a systematic exchange of messages) has long been sought that would allow invocation of the global set of objects in a fault-tolerant, but still efficient manner. The protocol would not have the number of messages proportional to the number of objects or, even worse, to the square of the number of objects. Those skilled in the art have heretofore been unsuccessful in creating such a protocol.

DISCLOSURE OF THE INVENTION

The present invention provides a hierarchical method for fault tolerance in a distributed computer system. A plurality of data centers is provided having a plurality of objects in each of the plurality of data centers. A local sub-protocol is used for dissemination of messages within a data center in the plurality of data centers and the local sub-protocol is activated from another data center of the plurality of data centers in a single round-trip message in the absence of faults.

The above and additional advantages of the present invention will become apparent to those skilled in the art from a reading of the following detailed description when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a basic (failure-free) interaction pattern of the protocol of the present invention in a distributed computer system;

FIG. 2 is an example of the behavior of the protocol of the present invention in the presence of a failure in a data center which is not the initiator data center; and

FIG. 3 is an example of the behavior of the protocol of the present invention in the presence of a failure in the data center, which is the initiator data center.

BEST MODE FOR CARRYING OUT THE INVENTION

The present invention uses a hierarchical method or protocol. Within each data center, a local sub-protocol ensures fault-tolerant dissemination of messages within that data center. This sub-protocol is then activated from another data center in a fault-tolerant manner, which only requires a single round-trip message if there are no failures. Essentially, the invention captures a trade-off between local-area and wide-area communication, where a few more messages are exchanged within a data center in order to reduce the number of messages that go between data centers.

The sub-protocol used within a data center uses an *atomic broadcast* protocol, which is a well-known building block for fault-tolerant systems. In addition to reliable message dissemination, an atomic broadcast protocol also ensures that different messages are delivered in the same order to all objects, such as replicated databases. The order property makes it more expensive to implement atomic broadcast (as compared to reliable broadcast). However, the order property allows the use of a primary-backup scheme within each data center. Only the current primary object within the initiator's data center communicates with other data centers. Thus, the election of a primary object enables satisfaction of a single-round-trip constraint.

Besides the availability of an atomic broadcast protocol within each data center, the protocol makes the following assumptions:

- *Failure detection.* The objects within a given data center have access to a failure detector that provides information about the failures of other objects in the same data center. It is assumed that the failure detector is eventually "strong". Roughly speaking this means that crashed objects are eventually permanently suspected to have crashed and eventually there is a correct object that is never suspected by another correct object. Failure detectors can make mistakes, that is, during certain periods of time, objects that have not crashed may be suspected to have crashed, and objects that have crashed are not suspected to have crashed.

- *Reliable channels.* It is assumed that every pair of objects is connected through reliable channels. That is, if an object sends a message to another object, and neither object crashes, then the message will eventually reach its destination.

Referring now to FIG. 1, therein is shown the basic (failure-free) interaction pattern of the protocol in a system 100 with three data centers 101, 102, and 103. The vertical lines

represent objects 111-119. An object 111 in the data center 101 wants to invoke all other objects using an initiator invocation 120. It does so by activating an atomic broadcast protocol 125, represented by an atomic broadcast box, within the data center 101. There is a primary object 113 within the data center 101, and this primary object 113 relays messages, such as propagation messages 121 and 122, to the other data centers 102 and 103. In the other data centers 102 and 103, the receiver of the propagation message activates the local atomic broadcast protocol, atomic broadcast protocols 126 and 127, to disseminate the message locally. When the atomic broadcast protocol 126 or 127 delivers the message to the receiver, it acknowledges receipt by sending a message, such as a message 123 or 124, to the primary object 113 in the initiator's data center 101.

In the figures, connector boxes 131-142 over the atomic broadcast protocol 125, 126, and 127 are used to indicate the use of the atomic broadcast. The connector boxes 131, 136, and 139 on top indicate that a process submits a message to the atomic broadcast system. The connector boxes 132-134, 135, 137-138, and 140-142 below the atomic broadcast box indicates that the broadcast system delivers a message to an object.

Circles are used to indicate the invocation of objects. An "X" circle is the actual invocation, such as invocations 150-159, and an empty circle is the request to invoke (generated by some object), such as the initiator invocation 120.

Referring now to FIG. 2, therein is shown an example of the behavior of the protocol in the presence of a failure of an object in a data center that is not responding to the initiator invocation 120, such as the object 115 of the data center 102, which has a crash 145.

The primary is the object 113 in the data center 101 (the initiator's data center) and it has chosen a default receiver, the object 115, as the object in data center 102 to receive communications. If the primary object 113 in data center 101 times out after waiting for an acknowledgement from the default receiver object 115, it simply selects another object in data center 102 to be the new receiver, e.g., a new receiver object, the object 114. It should be noted that the primary object 113 may suspect the default receiver object 115 to have crashed. This may be a false suspicions so the same message, such as an propagation message 144, may be sent to two or more objects in a given remote data center. To guarantee that each object is invoked once, it is necessary to keep track of such duplication. Standard techniques can be used for this. For example, a unique unit identifier (UUID) can be associated with

each message, and the receiver can then remember which messages has been received, and only use the same message for a single invocation.

Referring now to FIG. 3, therein is shown an example of the behavior of the protocol in the presence of a failure of a primary object in a data center that is responding to the initiator invocation 120.

The initial primary is the object 113. When the primary object 113 crashes 163, the object 111 detects the crash, for example through some timeout mechanism. When suspecting the crash at a time 165, the object 111 broadcasts a special message 167 that conveys this suspicion. The suspicions are also ordered, both with each other and with the normal messages, in the atomic broadcast facility. The ordering allows the objects in data center 101 to agree on the suspicion, and they can use a pre-determined, deterministic scheme to compute the next primary object. If the scheme is deterministic (e.g., round robin) they will agree on the identity of the next primary, for example, the new primary object could be the object 112. It should be noted that the suspicion may be false. For example, a network partition may have caused the original primary object 113 to appear to have crashed. The system will remain consistent even in that case because the original primary object 113 will then deliver a special suspicion message 169 and it will voluntarily cease to be primary and become a backup.

The new primary object 112 has to ensure that all messages that were supposed to be sent to other data centers by the original primary object 113 are in fact sent. One simple way to do this is for the new primary object 112 to send every message it has seen so far. A simple optimization of that naive scheme would be for the new primary object 112 to propagate an acknowledgement message from remote data centers 102 and 103 to the other objects 111 in the initiator's data center 101. If a message has been acknowledged in this way, a backup object can safely discard it: it is not necessary to for the initiator's data center 101 to send this message to the remote data centers 102 and 103 after becoming the new primary.

While the invention has been described in conjunction with a specific best mode, it is to be understood that many alternatives, modifications, and variations will be apparent to those skilled in the art in light of the aforegoing description. Accordingly, it is intended to embrace all such alternatives, modifications, and variations which fall within the spirit and scope of the included claims. All matters hither-to-fore set forth herein or shown in the accompanying drawings are to be interpreted in an illustrative and non-limiting sense.